Cognome e nome dello studente: ***Matricola:***

1. [5] Data la CPU di Figura 1, specificare il contenuto di **tutti** i registri (parte master e parte slave), quando è in esecuzione il seguente segmento di codice [5]:

```
0x0000 0400   addi $t3, $t1, 24
0x0000 0404   and $s0, $t1, $t1
0x0000 0408   sub $t1, $t2, $t3
0x0000 040C   lw $t1, 32($s0)
0x0000 0410   sw $t1, 64($s0)
```

quando l'istruzione di `addi` si trova in fase di WB. Specificare sullo schema (con colore o con tratto grosso) quali linee, all'interno dei diversi stadi, trasportino dati utili all'esecuzione dell'istruzione correntemente in quello stadio. Ci sono hazard nel codice precedente? Motivare la risposta. Modificare eventualmente la CPU in modo tale che questo codice venga eseguito correttamente.

2. [2] Modificare la CPU di Figura 1, in modo che possa gestire gli hazard sui dati anche nel caso di una istruzione di `beq`, ad esempio:

```
0x0000 0400   add $t1, $s0, $s1
0x0000 0404   beq $t1, $t0, 4
```

3. [5] Cosa sono gli interrupt e le eccezioni? Come vengono gestiti dalle architetture Intel e dalle architeture MIPS/ARM? Specificare gli elementi della CPU MIPS che sono dedicati alla gestione delle eccezioni e cosa contengono. Modificare la CPU di Figura 1 per potere gestire un'eccezione di "Miss". Cosa si intende per mascheramento degli interrupt? Viene praticato nei MIPS? Come vengono gestite le eccezioni e gli interrupt dai sistemi operativi sul MIPS? Scrivere uno scheletro di possibile codice.

4. [5] Modificare la pipeline in Figura 1 perché diventi una pipeline superscalare. Spiegare la ragione e lo scopo di **tutte le modifiche** più rilevanti da apportare ai diversi stadi. Che differenza c'è tra pipeline super-scalare e pipeline dotata di VLIW? Quali sono i vantaggi e gli svantaggi di un approccio rispetto all'altro. Qual è il migliore e perché?

5. [2] Come si implementa l'esecuzione vettoriale in una pipeline super-scalare? Mostrarlo modificando un cammino di esecuzione di una pipeline non dotata di esecuzione vettoriale.

6. [1] Spiegare chiaramente cosa si intende per stallo e illustrare almeno una tecnica per ottenerlo.

7. [4] Disegnare una memoria cache (parte dati + TAG + bit di validità – non è necessario disegnare le porte di lettura e scrittura) per un'architettura MIPS a 32 bit, a 4 vie di 256 KByte per banco, e linee di 8 parole (per ciascun banco). Definire cosa rappresenta il campo TAG e dimensionarlo opportunamente. Dove posso trovare il dato letto dall'istruzione lw $t1, 1024($0)? Da quanti bit è costituita questa memoria complessivamente? Cosa succede quando si verifica una miss? Come si può limitare la frequenza di miss? Da quale dei quattro banchi viene scaricato il dato quando occorre caricare una nuova linea? Perché?

8. [3] Cosa sono i codici di rilevamento e correzione degli errori? Come funziona il codice di Hamming? Calcolare il codice per il numero binario 1000 0000. Chi utilizza il codice di rilevamento e correzione degli errori? Schizzare uno schema a blocchi di un modulo, che mediante il codice di Hamming, possa correggere un errore singolo.

9. [2] Cosa rappresenta il "roof model"? Cosa rappresenta l'intensità aritmetica? Si riferisce ad una CPU o ad un particolare programma? Un programma che elabora matrici sparse sarà un programma con intensità aritmetica alta o bassa? Perché? Quali sono i passi per ottimizzare le prestazioni del codice suggeriti dal roof-model? Cos'è un kernel benchmark? Cos'è lo SPEC?

10. [5] Le memorie sono costituite da una gerarchia costruita secondo criteri ben precisi. Rispondere a queste domande motivando la risposta:
    a) Cosa si intende per gerarchia delle memorie?
    b) Spiegare chiaramente cosa si intenda per **coerenza** e **consistenza** di una memoria. Fare degli esempi. A quali memorie si applicano?
    c) Quali sono i meccanismi messi in atto per garantire la **coerenza** e la **consistenza** della memoria nelle architetture **mono-processore** e nelle architetture **multi-processore**? Quali sono i vantaggi e svantaggi di ciascun meccanismo?
    d) Cosa si intende per hit e miss di una meoria cache e come vengono gestiti? Chi li gestisce? Perché le miss sono critiche?
    e) Che differenza c'è tra una miss e un page fault? Cos'è un page fault?
    f) Cos'è la memoria virtuale? Cos'è la Tabella delle pagine? Dove si trova? Cos'è il "Translation Lookaside buffer"? Dove si trova? A cosa servono la memoria virtuale, il TLB e la tabella delle pagine?
    g) Che relazione c'è tra la memoria virtuale e la memoria fisica? Chi utilizza la memoria virtuale? Chi utilizza la memoria fisica? Cosa succede quando la CPU chiede una parola alla memoria?

11. [2] Riportare alcune caratteristiche della architetture x86.

Registri del register file

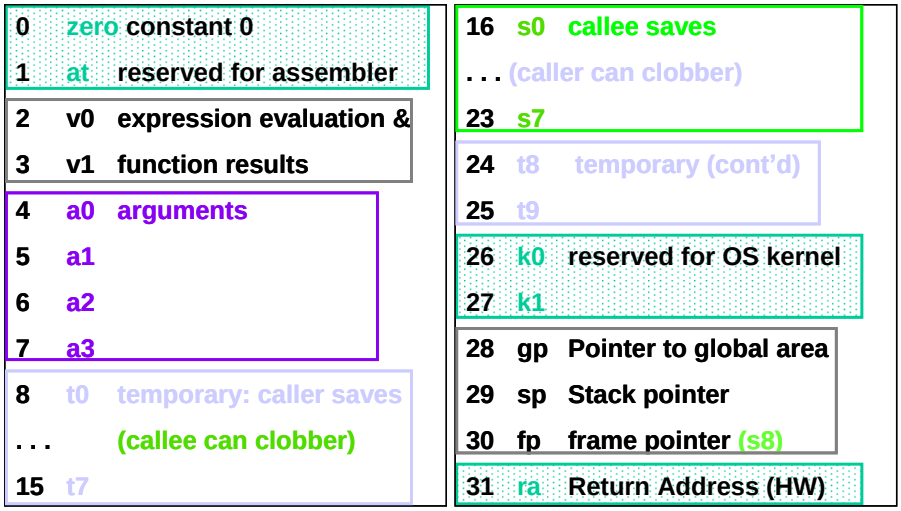
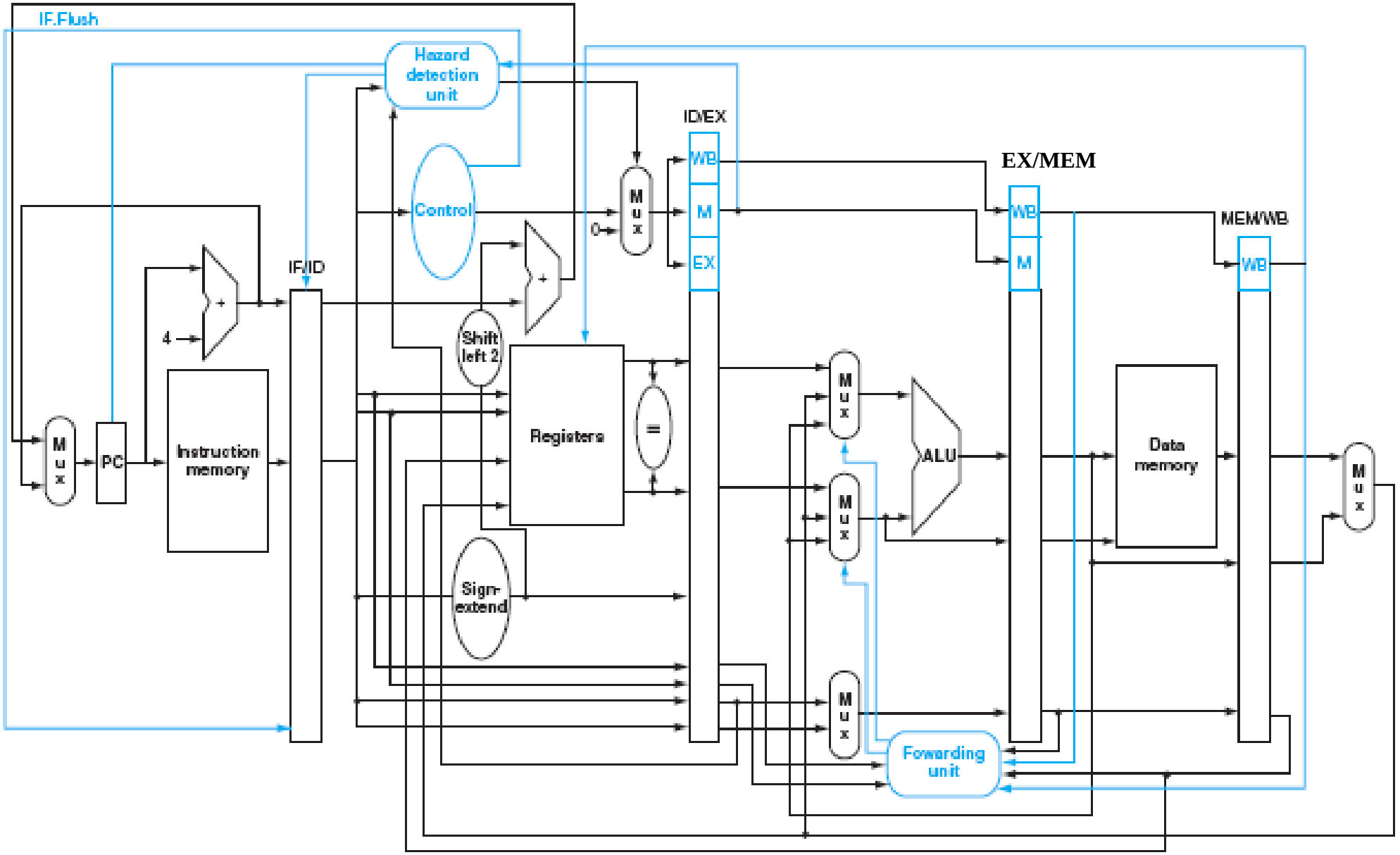


Figure 1


IF.Flush
Hazard detection unit
ID/EX
EX/MEM
MEM/WB
WB
M
EX
Control
M u x
0
IF/ID
+
4
Shift left 2
Registers
=
PC
Instruction memory
Sign-extend
ALU
Data memory
Fowarding unit